Geometric Reasoning in Sketch-based Volumetric Decomposition Framework for Hexahedral Meshing

Jean Hsiang-Chun Lu[1], Inho Song[1], William Roshan Quadros[2], and Kenji Shimada[1]

[1] Carnegie Mellon University, Pittsburgh, PA, hsiangcl@andrew.cmu.edu, {songphd, shimada}@cmu.edu
[2] Sandia National Laboratories†, Albuquerque, NM, wrquadr@sandia.gov

Abstract This paper presents a sketch-based volumetric decomposition framework using geometric reasoning to assist in hex meshing. The sketch-based user interface makes the framework user-friendly and intuitive; and the geometric reasoning engine makes the framework smarter and improves the usability. The system first creates a data structure containing B-Rep and 3D medial to capture the exterior and interior of the input model, respectively. The four-step geometric reasoning process consists of (1) Determining sweeping direction and two types of sweepable regions, (2) Providing visual aids on sweeping direction and sweepable region for decomposition, (3) Understanding user's intent by using prioritized B-Rep and medial entities, and (4) smart decomposition operation. Imprint and merge operations are then performed on the decomposed model before passing it to the sweeping algorithm to create hex meshes. The sketch-based framework has been tested on industrial models.

Keywords: 3D medial object, geometric reasoning, hexahedral meshing, sketch-based decomposition.

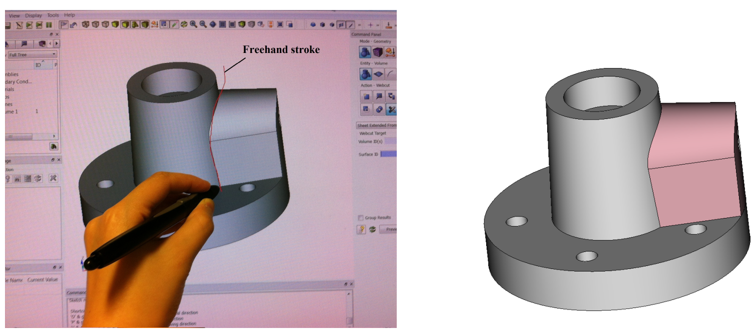
1 Introduction

Hexahedral meshes yields more accurate result in numerical analyses and are thus more desirable than tetrahedral meshes [1]. However, a hexahedral mesh is difficult to generate. To generate hex elements, volumetric decomposition is frequently conducted. The decomposition process is time consuming and

†Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

cumbersome to novice users because it requires special user expertise on volumetric decomposition and to be familiar with the operation of a CAE package to complete the task.

Automatic hex meshing methods have been presented by few researches [2–5], however, non hex-meshable sub-domains still remain and requires manual decomposition. The main bottleneck of manual decomposition is the existing user interface (UI) that requires detailed geometric information for cutting surface creation, and offers limited guidance for decomposition. A series of complicated actions must be conducted by selecting menus, icons, and type in parameters to define well-aligned cutting surfaces; and determining the ideal cutting regions requires user expertise. Lu et al. [6] presented a sketch-based decomposition method to improve the efficiency of decomposition by allowing using freehand strokes to create cutting surfaces (Fig. 1). Freeform cutting surface can be created easily, and the automatic snapping and alignment ensure the mesh quality. The sketch-based approach speeds up the cumbersome decomposition process by freeing the users from having to input details in traditional GUI or command line. This paper is an extension of the previous work, new improvements that enhances the performance of the sketch-based decomposition have been presented.



(a) (b)

Fig. 1: The sketch-based decomposition. (a) A freehand stroke is created in the sketch-based UI. (b) The stroke decomposed the model into two sub-domains.

In this paper, we propose a framework that uses intelligent geometric reasoning to improve the sketch-based decomposition task. The Geometric reasoning database contains B-Rep and 3D medial. The main five steps of the approach are: (1) Visualizing decomposition suggestions, (2) accepting sketch-based inputs from the user, (3) understanding the user's intents, (4) conducting smart decomposition, and (5) hex-meshing the model after conducting imprinting and merging.

One of the primary contributions of this paper is the geometric reasoning that brings the smartness to the sketch-based decomposition framework.

The proposed framework also improves the medial grouping algorithm [7] for sweepable region detection. The other contributions come from the new usage of 3D medial: (1) Determine sweeping direction and two types of sweepable regions. (2) Provide visual aids on sweeping direction and sweepable region for decomposition. (3) Determine ideal cutting position. (4) Prioritize B-Rep and medial entities for the smart decomposition operation. (5) Use touching curves (tangent points of the maximal ball) as snapping candidates for freehand strokes.

2 Related Work

2.1 Decomposition for hex meshing

Research has been done on decomposition based approach. Feature recognition techniques have been used for extracting manufacturing information from models [8–10]. Lu et al. [11] uses shape recognition techniques to extract decomposition features. Since the algorithm does not consider the sweepable volumes while detecting the decomposition features, it does not always result in hex-meshable sub-domains. Medial Axis Transformation (MAT) [12] has been applied on decomposition and meshing. Price at al. [13, 14] suggested using medial surface to guide decomposition and used the midpoint subdivision for meshing. However, the algorithm only works on certain classes of shapes. Shih et al. [15] generated swept volume for meshing. Heavily involved Boolean operations increase the computational cost of the algorithm. White et al. [16] automatically decompose multi-sweep volumes into many-to-one volumes by projecting the target faces through the volume onto corresponding source faces. This algorithm targets volume that are already multi-sweepable.

2.2 Sketch-based Decomposition

Sketch-based or pen-based approaches have been researched and applied in the computer aided design (CAD) field. The key concept is to create cutting surfaces by using strokes extracted from user's freehand input or existing drawings. These approaches improve the methods of inputting data and creating freeform surfaces in the CAD software or similar modeling systems. Igarashi et al. [17] proposed a sketch based system to create freeform 3D objects defined by closed strokes. Extrusion can be made on the objects. Varley et al. [18] converts a 2D sketch to B-Rep solid model insted of accepting direct stroke input from the user. Masry et al. [19] proposed optimization-based reconstruction algorithms to reconstruct sketches in a 3D sketching system for analysis. Kara et al. [20] presented a template-based approach for industrial design, which allows the user starting from modifying the templates to create 3D shapes.

Lu et al. [6] used a sketch-based UI to assist in geometric decomposition for hex meshing. The system takes user's freehand stroke as inputs to create the accurate cutting surfaces. The system evaluates the alignment type for the stroke to the existing boundaries. The stroke is then snapped and extruded to create the cutting surface. The well-aligned cutting surfaces prevent the bad angles between the boundaries, and ensure the mesh quality. However, the sweepable regions from the model are not considered during snapping candidate evaluation, and the position of cuts still requires user's expertise. A medial-based approach [7] has been proposed to improve the sketch-based decomposition. A 3D medial object is used to recognize the sweepable regions from the model. Each sweepable region is mapped to the model and visualized using different colors. The user then follows the visual aids and uses the strokes to perform the decomposition. In this approach, the strokes can only be snapped to existing boundaries, and some of the sweepable regions cannot be visualized if they are not bounded by existing boundaries. This paper is an extension of the previous work, and new improvments on the sketch-based decomposition and medial-based approach are presented.

3 Framework Overview

The key of sketch-based decomposition is to infer user's intent from the rough inputs for accurate decomposition instead of requiring detail geometry information to define cutting surfaces or decomposition operations. In this paper, the geometric reasoning brings the intelligence to the framework, and improves the sketch-based decomposition. The goal is to understand users intents from rough sketch-based inputs, and return a smart decomposition result. If the input is a freehand stroke, the geometric reasoning process searches for the best snapping candidate among the entities in the database for the input stroke. If the input is a series of picked entities, the reasoning process evaluates the possible operation for decomposition using those picked entities.

The sketch-based decomposition framework is shown in Fig. 2. After importing the model, a pre-processing (Sec. 4) creates the database for geometric reasoning. The sketch-based UI [6] in the front end accepts user inputs and supports user interaction. The geometric reasoning engine (Sec. 5) in the back end processes the inputs, reasons the database to understand users intent, and conducts a smart decomposition. After the decomposition, appropriate meshing algorithms are assigned to the sub-volumes to complete hex-meshing.

4 Pre-processing: Database Generation

The pre-processing step creates a database containing B-Rep and 3D medial for geometric reasoning. In a planar domain, medial is defined as the locus of

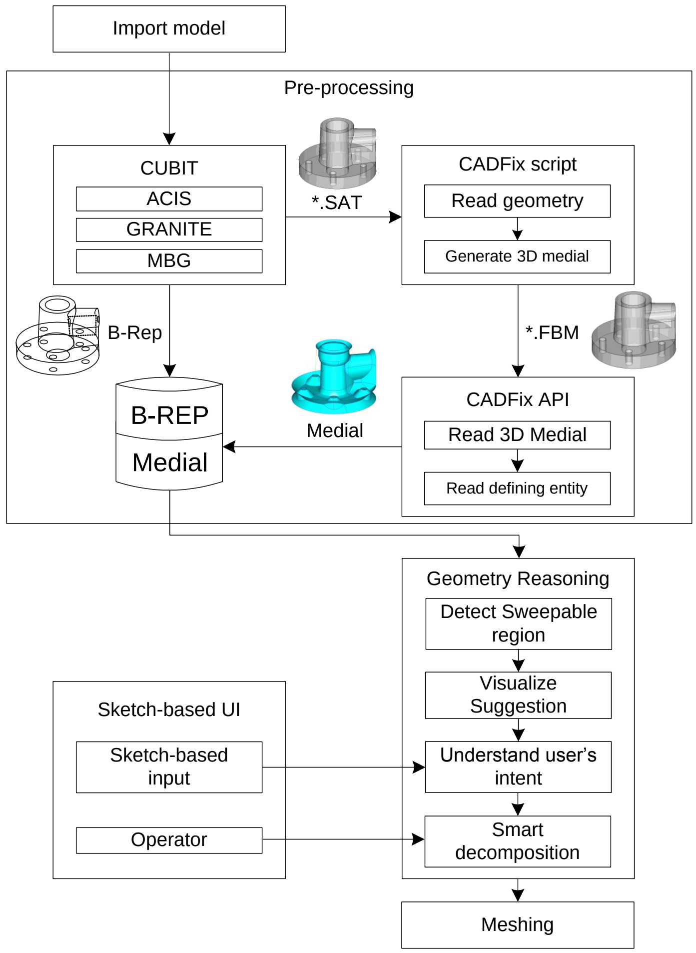
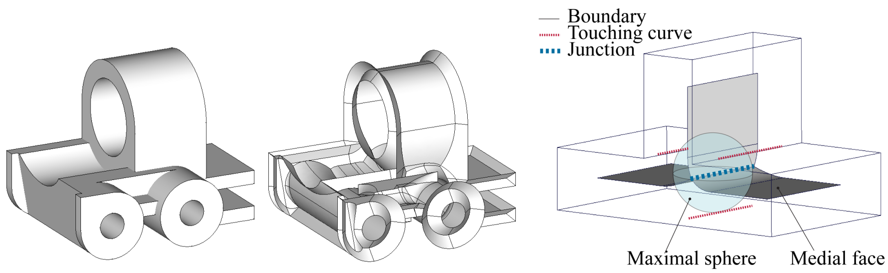


Fig. 2: The framework of the sketch-based decomposition with geometric reasoning.

the centre of the maximal ball as it rolls inside an object. It is a skeleton representation originally proposed by Blum [12]. The 3D equivalent (Fig. 3 (b)) is the locus of the center of maximal sphere. The medial related terminologies are listed as follows:

- Medial curve: a curve that connects two medial vertices.
- Medial face: a surface bounded by medial curves.
- Medial: a set of connected medial faces.
- Trimmed medial: the medial without the medial faces/curves that touch the model boundary.



(a) (b) (c)

Fig. 3: (a) The orignal model. (b) The 3D medial of the model. (c) The illustration of junction and touching curve.

- Medial segment: a set of 2-manifold medial faces connected by medial curves.
- Medial patch: a set of medial that is the sketleton representation of a sweepable region. Also called group of segments.
- Junction: A medial curve shared by more than two medial faces. (See Fig 3 (c))
- Defining entity: the original model's boundaries where the maximal sphere touches. These entities define the medial and hence are referred as defining entities.
- Touching site: The tangent point on the defining entitie where the maximal sphere touches. (See Fig 3 (c))
- Touching curve: A curve on the defining entity constructed by the touch sites of a junction.

The pre-processing is described as follows: First, the imported model will be assigned IDs to each entity. Next, we use a script to call CADFix [21], to generate the 3D medial. An application programming interface (API) is then used to obtain the 3D medial and establish the map between th e3D medial and the given model.

The design of the structure follows the concept of 3D medial as shown in Fig. 4. The structure has an "entity" class that contains ID, bound box, and centre points as class member. The entity class has "medial patch", "medial face", and "medial curve" as child classes.

A medial patch stores medial faces in the same patch, and their corresponding defining entity group. Given a medial face, its parent medial patch and child medial curve can be retrieved. Given a medial curve, its defining entity list and parent the medial face list can be retrieved. Each defining entity is assigned a unique ID in CUBIT. With the unique CUBIT ID, the defining entities can be mapped to the B-Rep. CUBIT uses the Common Geometry

Module (CGM) [22] to represent B-Rep solid model [23]. This way, given a medial entity, the B-Rep that defines the medial entity can be retrieved.

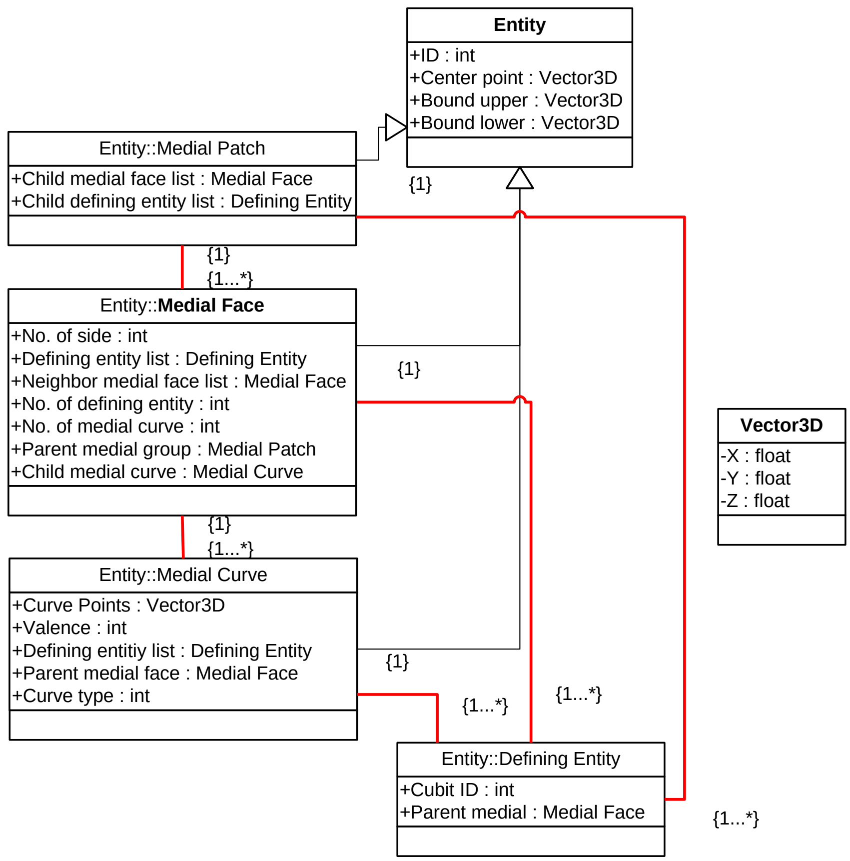


Fig. 4: The UML data structure to manage the medial information and the relationship between the B-Rep.

5 Geometric Reasoning for Sketch-based Decomposition

The goal of geometric reasoning is to figure out the desired cutting position, shape of the cutting surface, and the proper decomposition operation form users rough inputs. In the sketch-based UI, two types of inputs are accepted: freehand strokes, and picked entities. The freehand stroke is used to understand user's intent and then extruded to create a cutting surface. The stroke tells the desired shape of the cutting surface and the cutting position. The picked entities can be used to infer how the user wants to create cutting surfaces/decompose the model (e.g. sweep entity 1 along entity 2). The proposed geometric reasoning has four steps: Step 1: Detect sweepable region, Step 2: Visualize decomposition suggestions, Step 3: Understand user's intent, and Step 4: Smart decomposition.

5.1 Step 1: Detect Sweepable Region

The 3D medial could be used to infer the sweepable sub-domains. Lu et al. [7] presented a medial-based method to detect sweepable regions on trimmed medial by segmenting 2-manifold medial face, and grouping non-manifold medial faces that share the same end entities. The basic concept is that the 2-manifold segments are sweepable along the radius direction of the medial face patch using the defining entities as sweeping source and target surfaces (Fig. 5)(a), and the non-manifold groups are sweepable along the junction curve. However, the non-manifold groups that have the same end entities do not always have sweepable corresponding sub-domains on the given model. To make the grouping algorithm more accurate, this paper presented a new grouping method that checks number of curves on medial faces that are incident at a junction, and group those non-manifold medial faces. The resulting medial group has a corresponding sub-volume which is sweepable along the junction curve using the number of sides of a medial face.

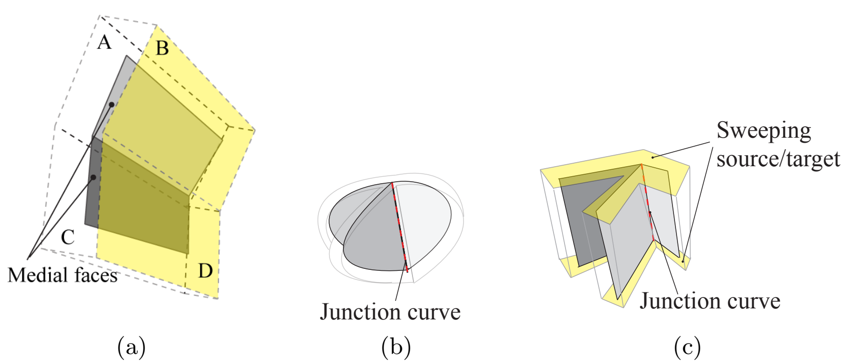


Fig. 5: Medial segments and groups. (a) Two medial faces that formed a patch represent their sweepable volumes can be united as one sweepable volume. Surfaces A, C are the sweep source and surface B, D are the target. (b) A group that contains two-sided medial faces. (c) A group that containts four-sided medial faces.

Grouping non-manifold medial patch

If a junction curve connects more than two 4-sided medial faces, the connected medial faces represented a sub-domain which is sweepable along the junction curve. In Fig. 5 (b) and (c), both cases have a junction curves shared by three medial faces. Medial faces in the first case are two-sided, and in the second case are four-sided. The corresponding volume in case shown in Fig. 5(b) is not sweepable along the junction curve because there is no linking surfaces; and the corresponding volume case Fig. 5(c) can be swept along the junction curve.

Combining the segmentation and grouping algorithms, the sweepable region detection process is as follows: (1) Segment 2-manifold medial segments. (2) Group the four-sided medial faces that share the same junction. (3) Map each medial patch to the corresponding sub-volumes on the model. Fig. 6 demonstrates the segmentatino and grouping process. The segmentation stops when encounters a junction as shown Fig. 6(d). The grouping algorithm detects a sweepable group as shown in Fig. 6(e). This process results in many medial patches (Fig. 6(f)) which have corrosponding sweepable regions on the model. Two types of sweepable regions based on the sweeping direction can be detected in this step: (1) sweep along the radius direction of the medial face, and (2) sweep along the junction curve.

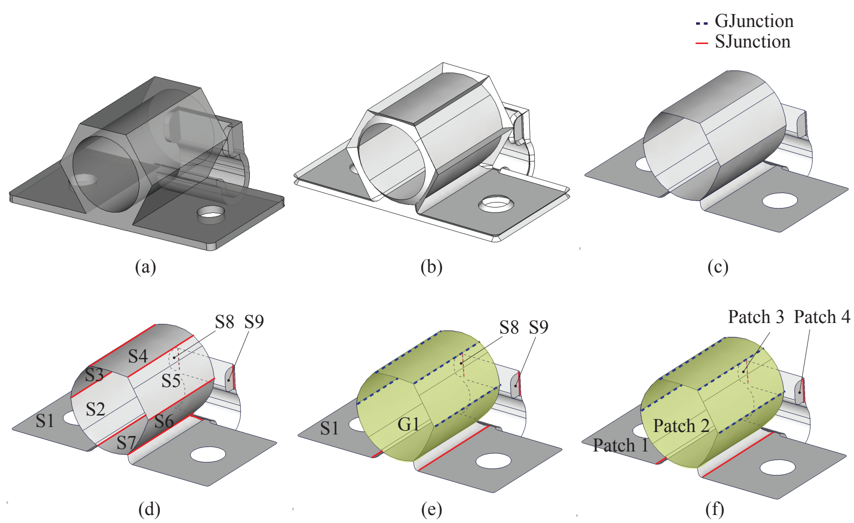


Fig. 6: (a) Original model (b) 3D medial. (c) Trimmed medial. (d) Ten 2-manifold segments (S1 to S9). (e) A non-manifold group (G1) obtaind by grouping S2 to S7. (f) Four medial patches are detected.

Types of junction

Two types of junctions can be defined based on the segmentation and grouping result (Fig. 7), and will be used in the next step:

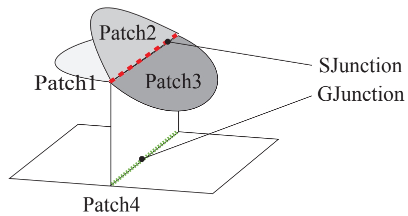


Fig. 7: Two types of junction on 3D medial illustration. Patches 1 to 3 are 2-manifold, and Patch 4 is non-manifold. Patches 1 to 3 are split by SJunction. Patches 3 is sweepable through its GJunction.

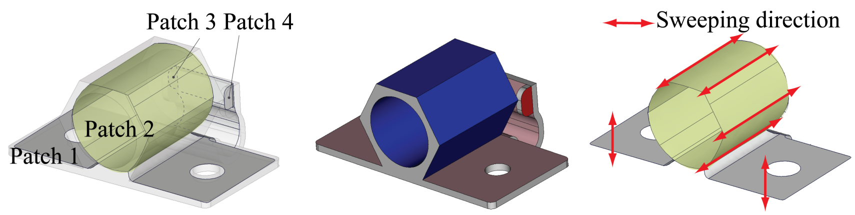
- SJunction: A junction that splits different medial segments or medial groups.
- GJunction: A junction in a non-manifold medial patch.

5.2 Step 2: Visualize Decomposition Suggestion

This step visualizes the decomposition suggestions generated by Step 1. If a medial patch does not contain any GJunctions, the patch has a corresponding sub-volumes sweepable along the radius direction of the child medial faces from the patch. If a medial patch contains a GJunction, the patch has a corresponding sub-volume sweepable along the GJunction. A model shown in Fig. 8 (a) contains four patches. Patches 1, 3, and 4 are medial segments, and Patch 2 is a medial group. The defining entities of each patch are color coded to visualize each sweepable region as shown in Fig. 8 (b). The suggested sweeping direction is displayed with arrows in Fig. 8 (c).

5.3 Step 3: Understand User's Intent

The users intent is to partition the model into sweepable sub-domains by creating cutting surfaces using strokes or existing entities. The goal of this step is to figure out where the user wants to cut and which entity the user wants to use for the decomposition operation. We thus have to prioritize the entities based on their potential to partition the sweepable regions.



(a) (b) (c)

Fig. 8: (a) Four medial patches are detectd. (b) The sweepable regions are visualized on the model. (c) The sweeping directions for the medial segment and group.

Entity prioritization

Two types of B-Rep and one type of medial entites have higher priority for decomposition:

- B-Rep entity: The defining surfaces of a medial segment.
- B-Rep entity: The defining edge of an SJunction.
- Medial entity: The touching sites of an SJunction.

For the medial segments which do not contain any GJunctions, the two defining surfaces of each child face are the sweeping source and target. Therefore, the defining faces have higher potential among all the other B-Reps to partition the sweepable sub-domains by extending themselves as a cutting surface.

In step 1, the 3D medial is used to detect sweepable regions: each medial segment and group has a corresponding sweepable volume, and is split by SJunctions. This means the touching sites of the SJunctions represent the ideal cutting position on the boundaries of the model. The touching sites could be a vertex (the tangent point of the maximal sphere) or a curve. If the curve matches an B-Rep edge, the edge is the defining edge of the SJunction. Otherwise, the curve is the touching curve formed by a series of tangent points.

An example of using a touching curve for decomposition is shown in Fig. 9. The model has two medial patches split by an SJunction. The touching curves shown in Fig. 9(b) are curves on the B-Rep that defines the SJunction, which indicate the partition line on the boundary. The 2D medial on the front surface is shown in Fig. 9(c). The touching site in the front view is the tangent point of the maximal circle. A cutting surface cut through the tangent points that is perpendicular to the boundaries is shown in Fig. 9(d). This makes the cutting area will always have ideal hex element. By avoiding the bad angles at the cutting area, mesh quality is ensured.

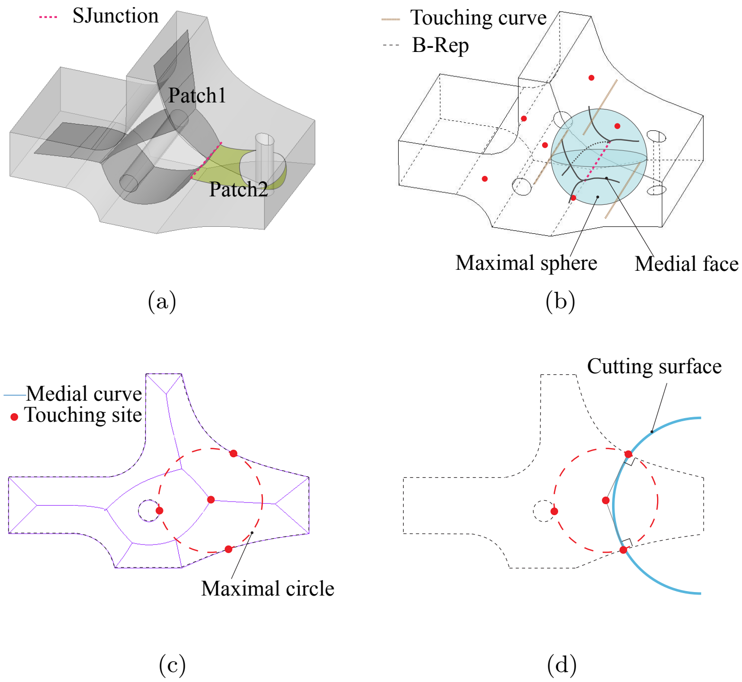


Fig. 9: (a) The original model contains two medial patches. (b) The illustration of SJunction's touching curves. (c) The front view of the medial and the B-Rep. The maximal sphere touches the boundary at three tangent points. (d) A cutting surface goes through the tangent points is perpendicular to the boundaries.

Stroke alignment and snapping

In order to use the freehand stroke on accurate decomposition, the stroke is snapped to B-Rep edges and medial touching curves. The sketch-based UI evaluates the alignment type using the method proposed in our previous paper on the pen-based UI [6]. The alignment types include offset, overlap, perpendicular and concentric. However, the alignment evaluation algorithm does not handle the gaps between the stroke's end points to the boundaries after snapping, which makes the stroke unable to cut through the body. Stroke extension and vertex snapping functions are provided to solve this problem. For a stroke $S(p_0, ..., p_n)$, if its end points are located on the model surface and are not connected to any boundaries after snapping, we first search if there are any B-Rep vertices near the end points within a pre-defined distance. If so, we snap the end point to the closest vertex. Otherwise, we use the intersections of the boundaries and $\overrightarrow{p_1 p_0}$ or $\overrightarrow{p_{n-1} p_n}$ as the new end points of the stroke.

5.4 Step 4: Perform a Smart Decomposition

This step intelligently determines the appropriate decomposition command. It detects the different purposes of the same type of input. The current implementation is able to create a cutting surface without manually specifying the operation by the following methods with the associated input type: (1) Extending a picked surface. (2) Sweep a picked surface along another picked curve. (3) Revolve a picked surface by the axis of another picked periodic curve. (4) Revolve a picked curve by the axis of another picked periodic curve. (5) Fit a surface on a closed loop if the loop curves are picked. The automatic selection of one of the five operations is based on the number and type of picked geometric entities. If a user selects only one surface, the picked surface will be extended and the volume will be decomposed. If two entities are picked with the second one periodic, the first picked entity is assigned as a profile, and revolved about the axis defined by the second picked entity. If one or more curves are picked, we first check if the curves form(s) a closed loop. If so, then the picked curves are used as the bounding curves to create a cutting surface.

6 Results and Discussion

Fig. 10 (a) shows a non hex-meshable volume. An arc cutting surface is the ideal cutting surface that subdivides the volume through the touching site of the SJunction as shown in 9 (d). As discussed in Sec. 5.3, a surface cutting through the tangent points (touching site of the SJunction) orthogonally ensures the generation of the ideal hex mesh at the cutting region. Fig. 10 (c) is the mesh generated using the decomposition solution obtained via geometric reasoning, and the hex elements highlighted in Fig. 10 (d) have a min Scaled Jacobian of 0.969 at the tangent points. The framework intelligently creates the cutting surface that does not deteriorate the mesh quality. Using random planar cutting surface (commonly used manual solution) shown in Fig. 10 (e) to produce sweepable sub-domains; however, the mesh quality at the cutting region has a min Scaled Jacobian of 0.746. An arc that is perpendicular to the boundaries could be created using the proposed sketch-based UI very easily, however, it does not guarantee a 90 degree intersection angle. When the geometric reasoning via a medial touching curve is used in combination with a user friendly sketch UI, the ideal cutting surface can be created.

Fig. 11 (a) shows an industrial non-sweepable model that requires decomposition for generation of hex mesh. The geometric reasoning detects the four medial patches. Patch 1 contains many GJunctions and it is sweepable along the GJunction direction as shown in Fig. 11 (b). Patches 1 to 3 are medial segments, and their defining entities are color coded on the model to display the sweepable region as shown in Fig. 11 (c). Following the decomposition suggestions, the user picks the surface, and the geometric reasoning engine

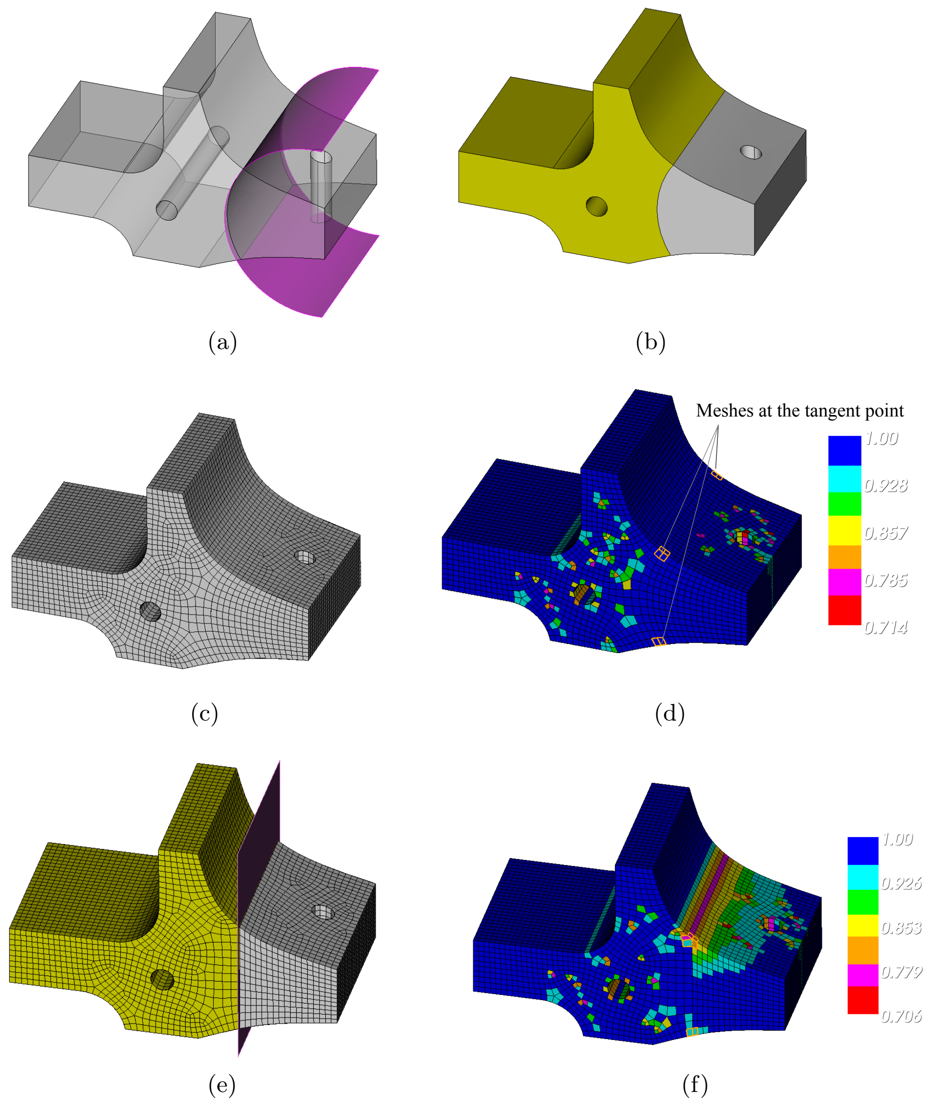


Fig. 10: (a) The cutting surface pass through the touching curve. (b) The decomposition result. (c) The all hex mesh. (d) The hex mesh quality using Scaled Jacobian. (e) Decompose the model with a planar surface. (f) The hex mesh quality using Scaled Jacobian.

determines the appropriate decomposition operation to extend the picked surface to sub-divide the model, and the sub-domains are generated. The model can be all hex-meshed after setting the sweeping direction for the grey volume, and performing imprinting and merging.

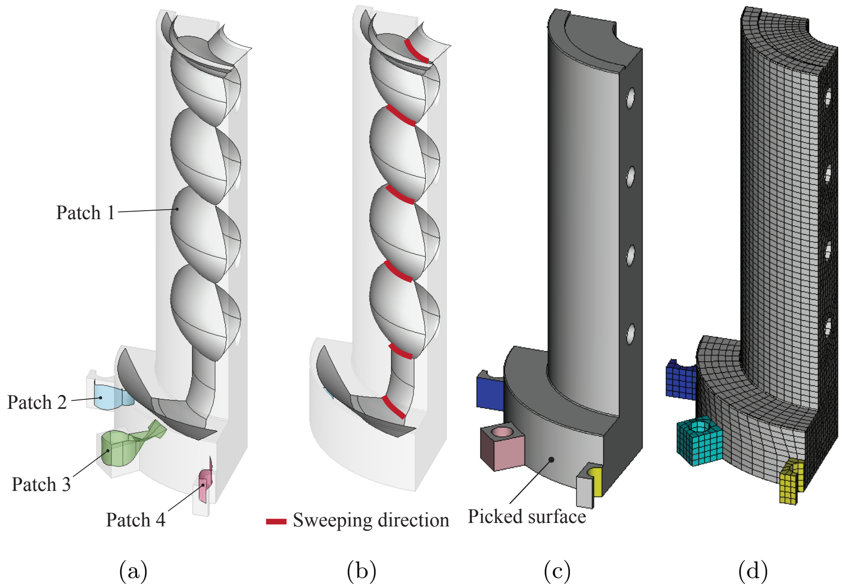


Fig. 11: (a) The medial patches marked on the trimmed medial. (b)The sweeping direction is visualized on the original model. (c) The sweepable region is visualized on the model with different colors. (d) The all hex-meshed output.

Fig. 12 (a) shows a fluid field around a turbine blade. Fig. 12 (b) shows that the 3D medial are split into many medial segments by SJunctions. After the medial grouping process, Fig. 12 (c) shows that the segments are grouped as one single patch, which is sweepable along the GJunction. This indicates the volume is sweepable in the same direction. When this volume is meshed, appropriate topology must to be chosen to ensure the mesh quality. In order to control the mesh size and orientation, the volume has to be decomposed into blocks which have appropriate topology for mapping/submapping, and meet CFD requirements. Using the proposed sketch-based UI, the volume can be decomposed following the pattern shown in Fig. 12 (d) using five freehand strokes. Note that the first stroke must cut through the whole volume to generate two sub-domains. When drawing the strokes, the end points are snapped to verties if they are close to corners. Therefore, we can obtain an accurate shape for the blocks. Fig. 12 (e) and (f) demonstrate the decomposition using stroke 5. After one more cut with stroke 6, seven sub-domains are generated as shown in Fig. 12 (g). By conducting imprinting and merging, the volume can be all hex-meshed as shown in Fig. 12 (h) by sweeping along the GJunction.

The geometric reasoning tells the volume is sweepable in the GJunction direction, and detects the corner vertex to snap the freehand stroke input. By using the proposed framework on this example, the block structure can be generated easily, and the mesh can be oriented along the block boundaries.

7 Conclusion

The paper presents a sketch-based volumetric decomposition framework using geometric reasoning to speed up the challenging and time-consuming decomposition process. The geometric reasoning approach infers users intent and returns an accurate decomposition from the rough sketch-based inputs. One of the main contributions is that the geometric reasoning brings the smartness aspect to the framework by a four-step approach: (1) detecting sweepable regions and sweeping direction; (2) providing visual aids for decomposition; (3) understanding users intent by prioritizing outer/B-Rep and skeletal/MO-based snapping candidates, determining ideal cutting position and alignment/snapping types; and (4) conducting smart decomposition. The proposed method has been tested on industrial models by generating hex meshes using sweeping algorithms.

Acknowledgement. The authors would like to thank Dr. Geoffrey Butlin, Mr. Henry Bucklow, Mr. Robin Fairey, Mr. Mark Gammon, Mr. Mike Field, and Mr. John Lamont for assisting with the medial related work in CADFIX.

References

1. S. Yamakawa, I. Gentilini, and K. Shimada, “Subdivision templates for converting a non-conformal hex-dominant mesh to a conformal hex-dominant mesh without pyramid elements,” *Engineering with Computers*, vol. 27, pp. 51–65, 2011.
2. T. J. Tautges, T. Blacker, and S. A. Mitchell, “The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes,” vol. 39, pp. 3327–3349, 1996.
3. N. T. Folwell and S. A. Mitchell, “Reliable whisker weaving via curve contraction,” in *Proceedings of the 7th International Meshing Roundtable*, pp. 365–378, 1998.
4. R. Schneiders, “Automatic generation of hexahedral finite element meshes,” in *Proceedings of the 4th International Meshing Roundtable*, pp. 103–114, 1995.
5. R. Schneiders, “A grid-based algorithm for the generation of hexahedral element meshes,” *Engineering with Computers*, vol. 12, pp. 168–177, 1996.
6. J. H.-C. Lu, I. H. Song, W. R. Quadros, and K. Shimada, “Pen-based user interface for geometric decomposition for hexahedral mesh generation,” in *Proceedings of the 19th International Meshing Roundtable*, pp. 263–278, 2010.

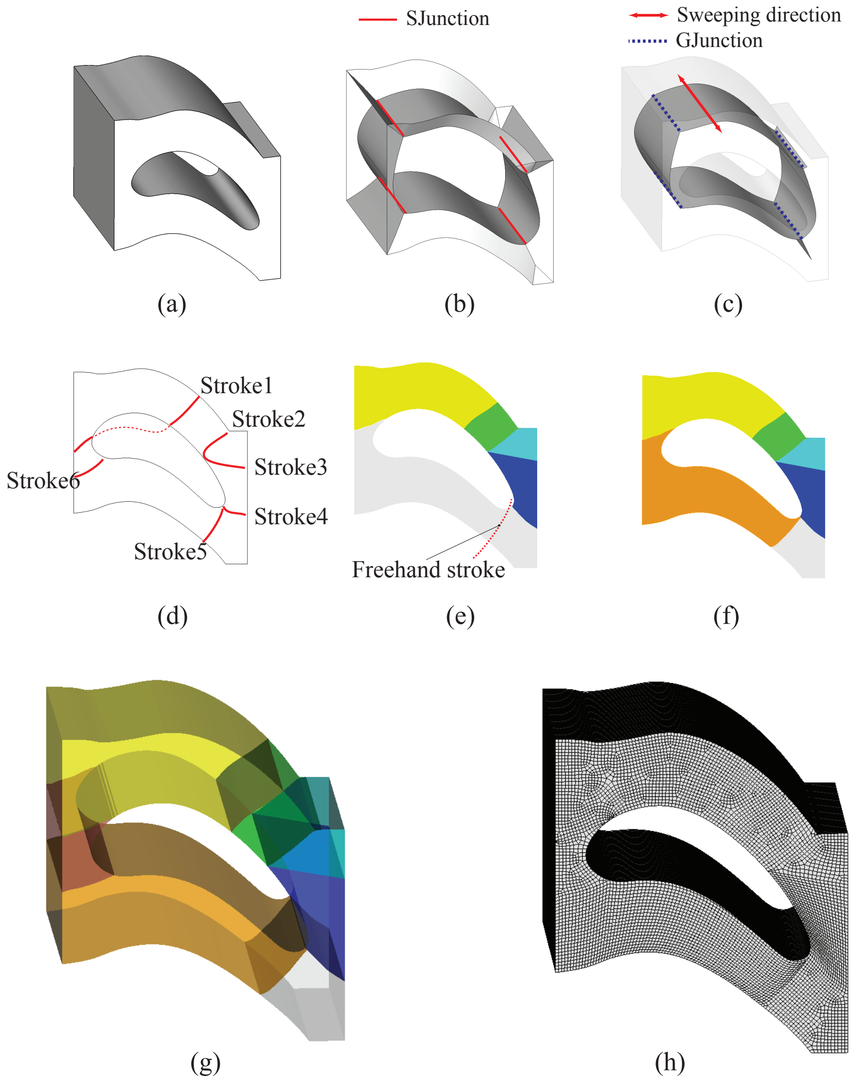


Fig. 12: (a) A flow field around an turbine. (Reproduced from [24]) (b) Four SJunctions split many medial segments. (c) The segments have been grouped into one single patch, which is sweepable along the GJunctions. (d) The illustration of strokes to decompose the model. (e) The user draws stroke 5. (f) The decomposition result after conducting five cuts. (g) The final decomposition result. (h) The all hex mesh output.

7. J. H.-C. Lu, I. H. Song, W. R. Quadros, and K. Shimada, "Volumetric decomposition via medial object and pen-based user interface for hexahedral mesh generation," in *Proceedings of the 20th International Meshing Roundtable*, pp. 179–196, 2011.
8. S. Subrahmanyam and M. Wozny, "An overview of automatic feature recognition techniques for computer-aidedprocessplanning," *Computers in Industry*, vol. 26, pp. 1–21, 1995.
9. B. Babic, N. Nesic, and Z. Miljkovic, "A review of automated feature recognition with rule-based pattern recognition," *Computers in Industry*, vol. 59, pp. 321–337, 2008.
10. L. Kyprianou, "Shape classification in computer aided design," in *PhD Thesis, University of Cambridge*, 1980.
11. Y. Lu, R. Gadh, and T. J. Tautges, "Feature based hex meshing methodology: feature recognition and volume decomposition," *Computer-Aided Design*, vol. 33, no. 3, pp. 221–232, 2001.
12. H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, pp. 362–380, 1967.
13. M. A. Price, C. G. Armstrong, and M. A. Sabin, "Hexahedral mesh generation by medial surface subdivision: part I. solids with convex edges," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 19, pp. 3335–3359, 1995.
14. M. A. Price and C. G. Armstrong, "Hexahedral mesh generation by medial surface subdivision: part II. solids with flat and concave edges," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 1, pp. 111–136, 1997.
15. B.-Y. Shih and H. Sakurai, "Automated hexahedral mesh generation by swept volume decomposition and recomposition," in *Proceeding of the 5th International Meshing Roundtable*, pp. 273–280, 1996.
16. D. R. White, S. Saigal, and S. J. Owen, "CCSweep: automatic decomposition of multi-sweep volumes," *Engineering with Computers*, vol. 20, pp. 222–236, 2004.
17. T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," in *Proceeding of the 26th annual conference on Computer graphics and interactive*, pp. 409–416, 1999.
18. P. A. C. Varley, H. Suzuki, J. Mitani, and R. R. Martin, "Shape classification in computer aided design," *International Journal of Shape Modeling*, vol. 6, 2000.
19. M. Masry, D. Kang, and H. Lipson, "A freehand sketching interface for progressive construction of 3D objects," *Computers and Graphics*, vol. 29, no. 4, pp. 563–575, 2005.
20. L. Kara and K. Shimada, "Construction and modification of 3D geometry using a sketch-based interface," in *Proceeding of the EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, pp. 59–66, 2006.
21. ITI TranscenData, "CAD Translation - CADFix," in $www.cadfix.com$.
22. T. Tautges, "CGM: a geometry interface for mesh generation, analysis and other application," *Engineering with Computers*, vol. 17, pp. 299–314, 2001.
23. L. Schoof and V. Yarberry, "Exodus II a finite element data model," *SAND92-2137, Sandia National Laboratories*, 1995.
24. Pointwise Inc., "Multi-block grids for axial turbines," in $www.pointwise.com/theconnector/March-2011/Gridding-an-Axial-Turbine-Video.shtml$, March 2011.